MEDIT   OCT. 12TH VERSION

The supplied diskette contains:

1. DOS 2S
2. MEDIT   (Mike's editor)
3. MFORMS   (Dale's forms program)

Notes on MEDIT

To use the editor, use the "L- binary load" command in DOS. The editor will sign-on and prompt for a filename to edit. If the file you specify does not exist, the editor will create it. If only a filename is given, "D:" is assumed. Be careful to check that there is enough free space on the disk to hold both a copy of the file you would like to edit and any additions you plan to make during the edit.

Note that a disk full error (error 162) is fatal in the editor. In fact, all I/O errors are FATAL in this version. You are advised to exit and re-enter the editor every 15 minutes (or so) of editing. That way if the system or editor dies you have only lost up to 15 minutes of work.

When errors or error warnings occur the keyboard speaker beeps and the error message is displayed in the error window. In this version the error message stays in the window until another error occurs or you delete a character in the command window. If errors occur that are internal to the editor (i.e. EDITOR IS CONFUSED) the keyboard ignores keystrokes until the secret key is typed. The secret key is CTRL. CLEAR. Usually a series of CTRL. CLEAR's will restore the editor. If not, try to EXIT or ABORT. If this type of error happens, PLEASE try to find a repeatable sequence that leads up to the error and tell me about it.

The editor should be as advertised in the Oct. 12th documentation; except:

- The CBF command should not be used.
- If an error occurs while executing the command line, the command line cursor is left in the command window.
- Delete-left (or Delete-before) does not work from the home position and should not be attempted.
- I/O errors are FATAL.
- Any others bugs are virgins and should be reported to me.

NOTE: using MEM.SAV while editting may crash BASIC.

Notes on MFORMS

MFORMS is my modified version of Dale's forms program version 2.2. This program formatted the documentation for the editor. To use this program type 'RUN "D:MFORMS' while in BASIC. The program asks for an input file; you supply the filename preceeded by "D:". The program then asks for an output file; you supply "P:" or a filename preceeded by "D:". If the output is sent to a file it can be later copied to the printer by using the DOS "C - copy to,from" command.

While using MEDIT to enter your text, I suggest you have each sentence followed by a space, CR combination. This allows for easy insertion or deletetion of text later. Also the formatter will put two spaces before starting a new sentence for better readability. Note that a blank line (two CR's in a row) in the input file flags the formatter to quit.

MEDIT - A SOURCE FILE EDITOR

PRELIMINARY VERSION -- 10/12/80

```
*******************************
*                             *
*  PLEASE REPORT ANY BUGS OR  *
*  ADVICE TO MIKE LORENZEN    *
*                             *
*******************************
```

INTRODUCTION

MEDIT is a source file editor. A source file is defined to be a disk file consisting of line(s) of ATASCI characters terminated by ATASCI "EOL"'s. The maximum length of any line as far as MEDIT is concerned is 199 columns. This value may be reduced by the user. Tabs can be treated the way the Operating System treats them (substituting n blanks to the next tabstop) or as expanding tabs. Expanding Tabs are tabs that only take one character in the file but are displayed as many columns of blanks. MEDIT uses no line numbers.

MEDIT divides the screen into three windows. The first window is the text window and is 20 lines long. The second window is the error window which is a single line long. The third window is the command window and is three lines long.

The text window displays up to 800 characters around the cursor. The window is updated instantly after every command to reflect the current state of the file.

The error window displays the last error that occured or prompts the user to answer MEDIT's questions.

The command window displays a single command line made of one or more commands. MEDIT allows more than one command line, but only displays the one you select.

MEDIT uses the common two-file method of editing. Text is copied from the file to be editted into the memory buffer where text can be added, deleted, or modified. When the memory buffer becomes full it is written to a temporary file. This process is repeated until all text has been copied from the editted file into the temporary file. Upon normal exit the .BAK file is deleted, the editted file is renamed as the new .BAK file, and the temporary file is renamed as the editted file. Upon abnormal exit the temporary file is deleted and the original editted file and .BAK are untouched.

IMMEDIATE MODE

Immediate mode refers to reserved keystrokes that immediately execute a command; any remaining keystrokes are inserted directly into the text at the current cursor position. By using the standard escape sequence any of the 256 possible characters can be inserted into the text. After any keystroke the text window is updated to reflect the current state of the file.

If the cursor is inside a expanding tab or to the right of a CR when a character is inserted the editor will automatically pad blanks to the left, insert the character, and move the tab to the right of the cursor.

IMMEDIATE MODE RESERVED KEYSTROKES

ctrl. left arrow	move cursor left (skip across tabs)
ctrl. right arrow	move cursor right (skip across tabs)
ctrl. down arrow	move cursor down one physical line
ctrl. up arrow	move cursor up one physical line
ctrl. 2	move cursor to beginning of logical line
ctrl. 3	move cursor to end of logical line
backspace	delete character left of cursor
ctrl. delete	delete character right of cursor
shift delete	delete logical line cursor is on
shift insert	prepare to insert a new line(s)
ctrl. clear	display prior screen-full of characters
ctrl. insert	display next screen-full of characters
tab	tab to next tab stop
shift tab	toggle "visible-tab-mode"
ctrl. tab	toggle "visible-cr-mode"
start	execute command(s) in command window
select	selects a different command line
option	erase command window-enter command mode
break	does nothing (NOP)
all other keystrokes	entered directly into text at cursor position

CTRL. LEFT ARROW - CURSOR LEFT

This command moves the cursor left one position except if the character to the left is a tab character while expanding tabs are enabled. If this exception is met the cursor will move completely over the expanded tab. If the cursor is moved above the text window an automatic scroll will pull down the previous line. To cursor left while already at the beginning of the text in the buffer will result in no operation except an error warning, "CURSOR AT END."

CTRL. RIGHT ARROW - CURSOR RIGHT

This command moves the cursor right one position except if the character to the right is a tab character while expanding tabs are enabled. If this exception is met the cursor will move completely over the expanded tab. If the cursor is moved below the text window an automatic scroll will pull up the next line. To cursor right while already at the end of the text in the buffer will result in no operation except an error warning, "CURSOR AT END."

CTRL. UP ARROW - CURSOR UP

This command moves the cursor up one physical line. If the cursor is moved above the text window an automatic scroll pulls down the previous line. To cursor up while the cursor is on the first line of the display window and no lines exist above the window causes the cursor to be moved to the beginning of the line and an error warning, "CURSOR AT END."

CTRL. DOWN ARROW - CURSOR DOWN

This command moves the cursor down one physical line. If the cursor is moved below the text window an automatic scroll pulls up the next line. To cursor down when the cursor is on the last line of the display window and no lines exist below the window causes the cursor to be moved past the last character in the buffer and an error warning, "CURSOR AT END" to be generated.

CTRL. 2 - MOVE THE CURSOR TO BEGINNING OF LOGICAL LINE

This command moves the cursor to the beginning of the logical line. If the cursor moves above the text window an automatic scroll occurs homing the cursor and the line. No errors can occur.

CTRL. 3 - MOVE THE CURSOR TO END OF LOGICAL LINE

This command moves the cursor to the end of the logical line. If the cursor moves below the text window an automatic scroll occurs pulling the rest of the line into the text window. No errors can occur.

BACKSPACE - DELETE LEFT OF CURSOR (IMMEDIATE MODE)

This command deletes the character to the left except possibly if the cursor is to the right of a carriage-return. In this case a cursor-left is done instead, unless the column to the left of the cursor is the CR. If a CR is deleted the two lines will be concatenated as long as the new line does not exceed the maximum column limits. If the maximum line length is exceded by this type of concatenation the deleted CR is restored, the command line is aborted with the error "LINE TOO LONG" leaving the user in immediate mode. If the cursor is in the middle of an expanding tab the entire tab is deleted. If the cursor is moved above the text window an automatic scroll sucks down the previous line. If the cursor is already at the beginning of the buffer no operation takes place except an error warning, "CURSOR AT END." It is illegal to delete the last CR in the buffer-if you try you will get the error warning, "CURSOR AT END". The only way to delete the last CR in the buffer is to use "Delete Line".

CTRL DELETE - DELETE CHARACTERS TO RIGHT OF CURSOR

This command deletes the character to the right except possibly if the cursor is to the right of a carriage-return. In this case a cursor-right is done instead, unless the column to the right begins the new line. If the cursor is in the middle of an expanding TAB the entire tab is deleted. If a CR is deleted the two lines will be concatenated as long as the new line does not exceed maximum column limits. If the maximum line length is exceded by this type of concatenation the deleted CR is restored, the command line is aborted with the error "LINE TOO LONG" leaving the user in immediate mode. If the cursor is already at the end of the text in the buffer no operation takes place except an error warning, "CURSOR AT END." It is illegal to delete the last CR in the buffer-if you try you will get the error warning, "CURSOR AT END". The only way to delete the last CR in the buffer is to use "Delete Line".

SHIFT DELETE - DELETE LOGICAL LINE CURSOR IS ON

This command deletes the logical line the cursor is on. The cursor is positioned before the first character of the next line. The last deleted line is stored in the "recover-line" buffer (see command RL).

SHIFT INSERT - PREPARE TO INSERT NEW LINE ABOVE CURRENT ONE

This command automatically moves the cursor to the beginning of the logical line, inserts a CR and moves the cursor before the newly inserted carraige-return. This allows a new line to be entered between two existing lines or a new line added past the last line in the buffer.

CTRL CLEAR - DISPLAY PREVIOUS SCREEN

This command displays the previous twenty physical lines above the text window and homes the cursor. If not enough lines exist above the window, the first twenty lines will be displayed and an error warning, "CURSOR AT END" is generated.

CTRL. INSERT - DISPLAY NEXT SCREEN

This command displays the next twenty physical lines below the text window and homes the cursor. If not enough lines exist below the window, the last nineteen lines will be displayed and an error warning, "CURSOR AT END", is generated.

TAB - INSERT A TAB INTO THE TEXT

If tabs have been selected to be like the Operating System Tabs the tab will be replaced by a number of blanks to move the cursor to the next tab stop. On the other hand, if tabs have been selected to be expanding tabs. The tab is also inserted into the text, but may take up more than one column.

SHIFT TAB - TOGGLE VISIBLE-CR-MODE

Carriage returns can either be displayed as blanks or down-arrows. This command switches to the opposite display character.

CTRL. TAB - TOGGLE VISIBLE-TAB-MODE

This keystroke only functions when expanding tabs have been selected. This command allows the expanding tabs to be displayed in either of two methods. These expanding tab fields can be displayed with blanks for easy viewing or as a right triangle followed by periods for easy editting. This command switches to the opposite display mode.

OPTION - ERASE COMMAND WINDOW - ENTER COMMAND MODE

This command erases the command window and enters the command mode. Following keystrokes will be inputted to the command window instead of the text window. Immediate mode commands, except backspace can still be executed from the command mode. The backspace instead will delete the last character typed into the command window.

START - EXECUTE WHATS IN THE COMMAND WINDOW

This command executes the command window if a syntactily correct command exists there. After execution the editor is left in Immediate-Mode. Executing a blank command or an empty display window will return control to the insert mode.

SELECT - SELECT ALTERNATE COMMAND LINE

Available soon.

COMMAND MODE

Command mode refers to trapping keystrokes for input into the command window instead of the text window. A mini-interpreter checks after each keystroke for valid syntax, thus helping the user to catch errors. Three kinds of Syntax errors can happen; "UNRECOGNIZED COMMAND", "DELIMETER ERROR", or "NUMBER TOO BIG." If a syntax errors occurs, the editor ignores keystrokes until the offending character is deleted from the command window.

The command window is three lines long, allowing a single command line made up of one or more commands. The OPTION key erases the entire command window, whereas Backspace (in command mode) deletes the last character entered into the command window. The START key will execute the command line if it is syntacticly correct and complete. After execution control is left in Immeditate mode. The command line is not erased after execution and may be re-executed by pressing START again.

CURRENTLY IMPLIMENTED COMMANDS

CLn	*	move cursor left n characters
CRn	*	move cursor right n characters
CUn	*	move cursor up n logical lines
CDn	*	move cursor down n logical lines
CBB	*	move cursor to beginning of buffer
CEB	*	move cursor to end of buffer
CBF	*	move cursor to beginning of file
CEF	*	move cursor to end of file
CBL	*	move cursor to beginning of logical line
CEL	*	move cursor to end of logical line
CCn	*	move cursor to column n
DLn	*	delete n logical lines
DBn	*	delete n characters before cursor
DAn	*	delete n characters after cursor
DR	*	delete remainder of logical line
IH	*	input half of available ram from file
IT/string/n	*	input string n times at cursor position
OC	*	output text to file up to line cursor is on
RL	*	recover previously deleted line
LMn	*	set left margin to width n
RMn	*	set right margin to width n
EXIT	*	exit normally from editor
ABORT	*	exit abnormally from editor nullifying all edits
(space)	*	used to make command lines readable

NOTE: "n" is an optional numeric argument that usually acts as a repeat counter with a range of 0-65535.

NOTE: "/string/" is a required character string delimited by either a pair of slashes or a pair of double quotes.

CLn - CURSOR LEFT n CHARACTERS

This command moves the cursor left n characters. If "n" is omitted or is zero the value is assumed to be one. This command cursors over tab fields and CR fields in a single bound. If the cursor is moved above the text window automatic scrolling pulls down the next line(s). To cursor left past the beginning of the buffer generates the error "CURSOR AT END" and aborts the command line, leaving the user in immediate mode.

CRn - CURSOR RIGHT n CHARACTERS

This command moves the cursor right n characters. If "n" is omitted or is zero the value is assumed to be one. This command cursors over tab fields and CR fields in a single bound. If the cursor is moved below the text window automatic scrolling pulls up the next line(s). To cursor right past the end of the buffer generates the error "CURSOR AT END" and aborts the command line, leaving the user in immediate mode.

CUn - CURSOR UP n LOGICAL LINES

This command moves the cursor to the beginning of the logical line that precedes the logical line the cursor is on. If "n" is omitted or is zero the value is assumed to be one. If the cursor is moved above the text window automatic scrolling pulls down the required number of lines. To cursor up past the beginning of the buffer generates the error "CURSOR AT END" and aborts the command line, leaving the user in immediate mode.

CDn - CURSOR DOWN n LOGICAL LINES

This command moves the cursor to the beginning of the logical line that follows the logical line the cursor is on. If "n" is omitted or is zero the value is assumed to be one. If the cursor is moved below the text window automatic scrolling pulls up the required number of lines. To cursor down past the end of the buffer generates the error "CURSOR AT END" and aborts the command line, leaving the user in immediate mode.

CBB - MOVE CURSOR TO BEGINNING OF BUFFER

This command moves the cursor before the first text line in the buffer. No errors can occur.

CEB - MOVE CURSOR TO END OF BUFFER

This command moves the cursor past the last line in the buffer. The screen is cleared leaving only the cursor. Usually the user would then use immediate "line-insert" to start typing additional text or use immediate "display prior 20 lines" to determine what text is at the end of the buffer. No errors can occur.

CBF - MOVE CURSOR TO BEGINNING OF FILE

This command moves the cursor before the first text line in the file.

CEF - MOVE CURSOR TO END OF FILE

This command moves the cursor past the last line in the file. The screen is cleared leaving only the cursor. Usually the user would use immediate "line-insert" to start typing additional text that would be appended to the end of the file.

CBL - MOVE CURSOR TO BEGINNING OF LOGICAL LINE

This command moves the cursor to the beginning of the logical line. If the cursor is already at the beginning of the line the command does nothing. No errors can occur.

CEL - MOVE CURSOR TO END OF LOGICAL LINE

This command moves the cursor to the end of the logical line. If the cursor is already at the end of the line the command does nothing. No errors can occur.

CCn - MOVE CURSOR TO COLUMN n

This command moves the cursor to the column specified by n. If the column specified is greater than the current length of the line but less than the maximum line length, the line is extended with blanks before the CR first. If n is greater than the maximum line length the command aborts with the error warning "NUMBER TOO BIG" and leaves the user in immediate mode.

DLn - DELETE n LOGICAL LINES

This command deletes the logical line the cursor is on. After deletion the cursor is placed before the first character of the next logical line. The last deleted line is stored in the "recover-line" buffer (see RL command). If "n" is omitted or is zero the value is assumed to be one. To execute this command when the cursor is past the last line in the buffer generates the error "CURSOR AT END" and aborts the command line, leaving the user in immediate mode.

DBn - DELETE n CHARACTERS BEFORE THE CURSOR

This command deletes n characters before the cursor. If "n" is omitted or is zero the value is assumed to be one. If a CR is deleted the editor will try to concatenate the lines. The new line must fit within current line length limits or the command line will abort with the error, "LINE TOO LONG" and the user is left in immediate mode. To delete characters before the cursor when the cursor is at the beginning of the buffer generates the error "CURSOR AT END" and aborts the command line, leaving the user in immediate mode. To delete the last CR in the buffer generates the error "CURSOR AT END" and aborts the command line, leaving the user in immediate mode. To delete the last CR in the buffer use the Delete-Line command.

DAn - DELETE n CHARACTERS AFTER THE CURSOR

This command deletes n characters after the cursor. If "n" is omitted or is zero the value is assumed to be one. If a CR is deleted the editor will try to concatenate the lines. The new line must fit within current line length limits or the command line will abort with the error, "LINE TOO LONG" and the user is left in immediate mode. To delete characters after the cursor when the cursor is at the end of the buffer generates the error "CURSOR AT END" and aborts the command line, leaving the user in immediate mode. To delete the last CR in the buffer generates the error "CURSOR AT END" and aborts the command line, leaving the user in immediate mode. To delete the last CR use the Delete-Line command.

DRn - DELETE REMAINDER OF LOGICAL LINE

This command deletes every character between the cursor location and the CR, except when the cursor is past the CR. If the cursor is past the CR the cursor is moved before the CR and nothing is deleted.

IH - INPUT HALF OF AVAILABLE RAM FROM FILE

This command calculates free memory and inputs approximately half that much from the file into the buffer. This command in combination with OC allows the user to edit files larger than will fit in memory. To IH when the INPUT EOF has already been reached does nothing except to generate an error warning, "INPUT EOF." If too little free memory exists the error "MEMORY FULL" is generated and the command line execution is aborted, leaving the user in immediate mode. Use OC to free up memory.

IT/string/n - INPUT STRING AT CURSOR POSITION n TIMES

This command inserts the string argument at the cursor location n times. If n is omitted or is zero, the value is assumed to be one. If the cursor is past the last line in the buffer an automatic insertion of a CR to the right of the cursor is done before inserting the text. If too little free memory exists either none or part of the string will be inserted and the error "MEMORY FULL" will be generated, aborting the command line execution, and leaving the user in immediate mode. The cursor is left after the last character inserted.

OC - OUTPUT TEXT UP TO THE CURSOR

This command outputs text from the beginning of the buffer up to the logical line the cursor is on; then the line the cursor is on becomes the first line in the buffer. This command in combination with IH allows the user to edit files larger than will fit in memory. This command frees up memory.

RL - RECOVER DELETED LINE

This command inserts the text line stored in the "recover-line" buffer in front of the current line the cursor is on. This is used both to recover from accidently deleting a line or as a simple way to move one line. Text is inserted into the recover-line buffer by either a command or an immediate keystroke to delete-logical-line.

LMn - LEFT MARGIN SET

This command changes the left margin. This is used in case your television is out of adjustment and cuts off columns on the left side of the display. The left margin may be set from 0 to 5 column via the n. If "n" is omitted it is assumed to be 0. If "n" is greater than 5 the command is aborted with the error warning "NUMBER TOO BIG", and leaves the user in immediate mode.

RMn - RIGHT MARGIN SET

This command changes the right margin. This is used in case your television is out of adjustment and cuts off columns on the right side of the display. The right margin may be set from 0 to 5 columns via the n. If "n" is omitted it is assumed to be 0. If "n" is greater than 5 the command is aborted with the error warning "NUMBER TOO BIG", and leaves the user in immediate mode.

EXIT - EXIT NORMALLY FROM EDITOR

This command is the normal exit from the editor. The buffer is written and filled repeatedly until the remainder of text is transfered to the output file. Then the old .BAK file is deleted, the input file is renamed as the new .BAK file, and the output file is renamed as the new file. The editor now being done returns to DOS.

ABORT - EXIT ABNORMALLY FROM THE EDITOR

This command is the abormal exit from the editor. WARNING - this command nullifies all edits done during the current editing session. The output file is closed and then deleted, leaving the original file and .BAK untouched. The editor now being done returns to DOS.

(space) - NO OPERATION

This command does nothing. It is usefull to sprinkle in between commands when more than one command is placed in the command window to make the command window more readable.

PROPOSED FUTURE COMMANDS

```
******************************************************************
SB/-/n         * Search through buffer for nth occurence of string
SF/-/n         * Search through file for nth occurence of string
SRB/-/-/n      * Search and replace n times in buffer
RF/-/-/n       * Search and replace n times in file
SRVB/-/-/n     * Search, replace and verify n times in buffer
SRVF/-/-/n     * Search, replace and verify n times in file
               *
               *
MC             * Clear block-markers from buffer
MS             * Insert a block-marker
BC             * Block copy to before logical line cursor is on
BM             * Block move to before logical line cursor is on
BD             * Block delete
BP             * Block print on system printer
BW/-/n         * Block write or write-out command window n
               * (can delete file too)
BR/-/n         * Block read or read-in command window n
               *
               *
REOPEN         * Re-open editor with same file
EXIT1          * Exit from edit, but remain in editor
ABORT1         * Abort from edit, but remain in editor
IN/-/          * Input from new file
ON/-/          * Output to new file
OA/-/          * Output to new file in append mode
PLn            * Print n lines on system printer
CTSn           * Convert expanding tabs to spaces for n lines
EWn            * Execute window n (macro)
RE             * Recover error
DF             * Delete first part of line up to cursor
               *
               *
Repeat groups  * Allow repeat groups in command lines
```

PRELIMINARY FORMS REFERENCE MANUAL, REV 2.2

By - Dale Yocum, 6/26/80

FORMS is an internal ATARI program which allows the user to produce formatted text on the ATARI 800. The user prepares his text using either the editior in the assembler or the disk based editior.

Anyone who is familiar with the DEC program RUNOFF or any of it's decendents will have no problem using FORMS.

TEXT FILE FORMAT

The text file, created using a seperate editior, contains both the textual material plus special commands to the FORMS program. A line beginning with a period is assumed to be a command and must match one of those listed in the Command Summary.

Usually the text is filled as it is processed. That is, the program fills a line by adding successive words from the text file until one more word would cause the right margin to be exceeded. Some commands cause a break in the output. A break means that no more filling will occure on the current line and the line is output to the printer. This occures, for example, at the end of a paragraph.

Note: only one command can appear per line.

COMMAND SUMMARY

.Ctttttt...

Center text. Causes a break after which the text tttttt... is centered on the following line. tttt... can be no longer than the right-hand margin set up by .Wn.

.D

Don't fill lines. Causes a break after which the source text is printed verbatim.

.D0

Turn on fill mode again.

.Fn

Skip n lines between pages. If n = 0 then issue a form-feed character between pages instead of just blank lines. Default is 8 lines between pages. There is o upper limit on the value of n.

.Hn

Page height. Changes the number of lines per page to n. Value of n can range from 5 to any value. Default is 57 lines.

.In

Indent lines. Causes a break after which the left margin in moved to the character position given by n. The right margin will remain constant. The value of n can range from 1 to the maximum line width (specified by .Wn)-1 Default value is 1.

.L

New line. Causes a break
You may accomplish the same function by placing a comma as the first character of a line of text. For example:

```
,This line will have a break.
```

Will do the same thing as:

```
.L
This line will have a break.
```

.Ln

Skip lines. Causes a break followed by n blank lines. The value of n can range from 1 to any value.

.Nn

Page number. Changes the number of the CURENT page to be n. If n is set to 0 then the printing of page numbers at the top of each page will be supressed. Default is to turn on page number printing. The first page is page 1. There is no maximum value for n.

.P

New paragraph. Causes a break after which one line will be skipped. The next line printed will be indented by the number of characters specified by .Pn below. Default is 5 characters.

.Pn

New paragraph and change. Causes a break after which a new paragraph will be created as described in .P above. The number of characters indented for this and all subsequent paragraphs is given by n. The value of n can range from 1 to the maximum line width - 1. The default value is 5 characters.

.Q

Quit at end of page. The output process will pause at the end of each page to allow the paper to be changed in the printer. Printing will resume when RETURN is pressed. This option will remain in force until turned off by .Q0 below.

.Q0

Don't quit at end of page. Turns off the pause at the end of each page. This is the default.

.Rttttttt...

Remark. The line ttttt... will only appear in the source text file. It will not be printed in the final output. There is no limit on the length of the remark string.

.S

Skip to top of page. Causes a break after which the system will leave the rest of the current page blank and advance to the top of the next page.

.Ttttt...

Title. The string ttttt... will be printed at the top of each subsequent page. The title can be up to 80 characters long. It is printed two characters after the page number. Page numbers must be enabled to print a title. Default is to print nothing.

.Vn

ertical spacing. Causes a break after which the line spacing is changed to n. For example: .V2 will double space the output. Default is single spacing.

.Wn

Maximum line width. Causes a break after which the right-hand margin is changed to n. This is the character position beyond which nothing will be printed. Default value is 78. The value of n can range from 1 to any value.

```
10 .R This is the source text for the
20 .R FORMS manual as processed by same.
30 .R Not all commands are used in this
40 .R text, but it should get the idea
50 .R across. This text was created
60 .R using the EDITIOR/ASSEMBLER
70 .R cartrige. FORMS version 1.X will
80 .R remove the line numbers for you.
90 .R
0100 .TFORMS Reference Manual
0110 .CPRELIMINARY FORMS REFERENCE MANUAL
0120 .CBy - Dale Yocum, 4/18/80
0130 .L3
0140 .P
0150 FORMS is an internal ATARI program
0160 which allows the user to produce
0170 formatted text on the ATARI 800.
0180 The user prepares his text using
0190 either the editior in the
0200 assembler or the disk based
0210 editior.
0220 .P
0230 Anyone who is familiar with the
0240 DEC program RUNOFF or any of it's
0250 decendents will have no problem
0260 using FORMS.
0270 .L2
0280 .CTEXT FILE FORMAT
0290 .P
0300 The text file, created using a seperate editior, contains both the textual
0310 material plus special commands to
0320 the FORMS program. A line beginning
0330 with a period is assumed to be a
0340 command and must match one of those
0350 listed in the Command Summary.
0360 .P
0370 Usually the text is filled as it
0380 is processed. That is, the program
0390 fills a line by adding successive
0400 words from the text file until one
0410 more word would cause the right
0420 margin to be exceeded. Some commands
0430 cause a break in the output. A break
0440 means
0450 that no more filling will occure
0460 on the current line and the line is output to the printer.
0470 This occures, for example, at the end
0480 of a paragraph.
0490 .P
0500 Note: only one command can appear per line.
0510 .L3
0520 .CCOMMAND SUMMARY
0530 .L2
0540  .Ctttttt...
0550 .L
0560 Center text. Causes a break after which
0570 the text tttttt... is centered on the following line.
0580 tttt... can be no longer than the right-hand margin set up by .Wn.
0590 .L2
0600  .D
0610 .L
0620 Don't fill lines. Causes a break after which
```

* Forms only recogni
a period in col. 1.
This will be printed as i
not as a command.

```
0640 .L2
0650  .D0
0660 .L
0670 Turn on fill mode again.
0680 .L2
0690  .Fn
0700 .L
0710 Skip n lines between pages. If n = 0
0720 then issue a form-feed character between
0730 pages instead of just blank lines.
0740 Default is 8 lines between pages.
0750 There is no upper limit on the value of n.
0760 .S
0770  .Hn
0780 .L
0790 Page height. Changes the number of
0800 lines per page to n. Value of n can
0810 range from 5 to any value. Default
0820 is 57 lines.
0830 .L2
0840  .In
0850 .L
0860 Indent lines. Causes a break after which
0870 the left margin in moved to the character
0880 position given by n. The right margin
0890 will remain constant.
0900 The value of n can range from 1 to
0910 the maximum line width (specified by .Wn)-1
0920 Default value is 1.
0930 .L2
0940  .L
0950 .L
0960 New line. Causes a break
0970 .L2
0980  .Ln
0990 .L
1000 Skip lines. Causes a break followed
1010 by n blank lines. The value of n
1020 can range from 1 to any value.
1030 .L2
1040  .Nn
1050 .L
1060 Page number. Changes the number of
1070 the CURENT page to be n. If n is set
1080 to 0 then the printing of page
1090 numbers at the top of each page will
1100 be supressed.
1110 Default is to turn on page number
1120 printing. The first page is page 1.
1130 There is no maximum value for n.
1140 .L2
1150  .P
1160 .L
1170 New paragraph. Causes a break after
1180 which one line will be skipped.
1190 The next line printed will be indented
1200 by the number of characters specified
1210 by .Pn below. Default is 5 characters.
1220 .L2
1230  .Pn
1240 .L
1250 New paragraph and change. Causes a
1260 break after which a new paragraph
1270 will be created as described in .P
1280 above. The number of characters indented
```

```
1300 is given by n. The value of n can
1310 range from 1 to the maximum line width - 1.
1320 The default value is 5 characters.
1330 .L2
1340  .Q
1350 .L
1360 Quit at end of page. The output
1370 process will pause at the end of each
1380 page to allow the paper to be changed
1390 in the printer. Printing will resume
1400 when RETURN is pressed. This option
1410 will remain in force until turned
1420 off by .Q0 below.
1430 .L2
1440  .Q0
1450 .L
1460 Don't quit at end of page. Turns
1470 off the pause at the end of each page.
1480 This is the default.
1490 .L2
1500  .Rttttttt...
1510 .L
1520 Remark. The line ttttt... will only
1530 appear in the source text file.
1540 It will not be printed in the
1550 final output. There is no limit on
1560 the length of the remark string.
1570 .L2
1580  .S
1590 .L
1600 Skip to top of page. Causes a break after which
1610 the system will leave the rest of
1620 the current page blank and advance
1630 to the top of the next page.
1640 .L2
1650  .Ttttt...
1660 .L
1670 Title. The string ttttt... will be
1680 printed at the top of each subsequent
1690 page. The title can be up to 80 characters
1700 long. It is printed two characters
1710 after the page number. Page numbers
1720 must be enabled to print a title.
1730 Default is to print nothing.
1740 .L2
1750  .Vn
1760 .L
1770 Vertical spacing. Causes a break after
1780 which the line spacing is changed
1790 to n. For example: .V2 will double
1800 space the output.
1810 Default is single spacing.
1820 .L2
1830  .Wn
1840 .L
1850 Maximum line width. Causes a break
1860 after which the right-hand margin
1870 is changed to n. This is the character
1880 position beyond which nothing
1890 will be printed. Default value
1900 is 78.
1910 The value of n can range from 1
1920 to any value.
```